Extending Modern PaaS Clouds with BSP to Execute Legacy MPI Applications

Hiranya Jayathilaka, Michael Agun
Department of Computer Science
UC Santa Barbara, Santa Barbara, CA, USA

Abstract

As the popularity of cloud computing continues to increase, a significant amount of legacy code implemented using older parallel computing standards is outdated and left behind. This forces the organizations to port the old applications into new cloud platforms. This, however, violates the "develop once - run anywhere" principle promised by utility computing. As a solution to this problem, we explore the possibility of executing unmodified MPI applications over a modern parallel computing platform. Using BSP as a bridging model between MPI and the Hadoop framework, we implement a prototype MPI runtime for today's computing clouds, which eliminates the overhead of porting legacy code.

1 Introduction

Our main goal is to execute virtually any Message Passing Interface (MPI) [5] based C program on Hadoop [1], without making any changes to the MPI code or the implementation of the Hadoop platform. This type of transparency is crucial when migrating legacy code to modern cloud environments, although it may incur a performance penalty. To achieve this goal, we deploy a Bulk Synchronous Parallel (BSP) [7] overlay (Apache Hama [3]) on Hadoop. This doesn't require any changes to the Hadoop implementation or the configuration. Then we define a mapping from native MPI constructs to the BSP constructs so that MPI operations can be executed on the BSP overlay.

SOCC '13, Oct 01-03 2013, Santa Clara, CA, USA.
ACM 978-1-4503-2428-1/13/10.
http://dx.doi.org/10.1145/2523616.2525942

2 Design and Implementation

Our architecture consists of 3 main components – a custom MPI C library, a BSP job that coordinates the execution of MPI code and a collection of MPI tools for the Hadoop environment. User's MPI C code should be linked with our custom MPI C library. This library is responsible for intercepting MPI procedure calls and delegating them to the underlying BSP framework.

The BSP job uploads the binary executable of the user's MPI code into HDFS, and starts a number of BSP processes (tasks). These BSP tasks download the MPI code from HDFS, and run them as separate child processes. Whenever a child MPI process calls a MPI function, it is dispatched to our custom MPI C library, which makes a TCP call to the parent BSP process. The parent BSP process executes the function call on behalf of the child process using native BSP constructs.

We also provide two MPI tools, mpicc and mpirun, that can be used to transparently compile and run MPI C code on Hadoop.

3 Results and Conclusion

We tested our prototype on multiple applications (PI calculation and matrix multiplication), using the most common MPI primitives, on a small cluster of machines. We also compared our results against another MPI-to-Hadoop adapter which uses MapReduce [4] as the underlying bridging model [6]. We were able to run a variety of unmodified MPI codes using our implementation, and our test results show an acceptable level of performance. Our results confirm that BSP is a more flexible and performant model for MPI-to-Hadoop bridging.

Currently, other solutions are being developed to allow running MPI directly on Hadoop (e.g. YARN [2]). However, by implementing a lightweight adapter such as ours, MPI code can be deployed in the cloud today, without upgrading existing Hadoop clusters. We have also reduced the performance overhead by selecting BSP, which matches the MPI primitives closely.

Acknowledgements

This work was funded in part by Google, IBM, NSF grants CNS-0546737, CNS-0905237, CNS-1218808, and NIH grant 1R01EB014877-01.

References

[1] Apache Hadoop. `http://hadoop.apache.org`, 2013.

[2] Apache Hadoop NextGen MapReduce (YARN). `http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html`, 2013.

[3] Apache Hama. `http://hama.apache.org`, 2013.

[4] J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, Jan. 2008.

[5] Message Passing Interface Standard. `http://www.mcs.anl.gov/research/projects/mpi`, 2013.

[6] J. Slawinski and V. Sunderam. Adapting MPI to MapReduce PaaS Clouds: An Experiment in Cross-Paradigm Execution. In *Utility and Cloud Computing (UCC), 2012 IEEE Fifth International Conference on*, pages 199–203, 2012.

[7] L. G. Valiant. A bridging model for parallel computation. *Commun. ACM*, 33(8):103–111, Aug. 1990.